(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 May 2001 (25.05.2001)

PCT

(10) International Publication Number
WO 01/37193 A1

(51) International Patent Classification[7]: G06F 19/00

(21) International Application Number: PCT/US00/28002

(22) International Filing Date: 11 October 2000 (11.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/438,846 12 November 1999 (12.11.1999) US

(71) Applicant: NET PERCEPTIONS, INC. [US/US]; 7901 Flying Cloud Drive, Eden Prairie, MN 55344 (US).

(72) Inventors: SARWAR, Badrul, M.; 1025 29th Avenue S.E., Apt. F, Minneapolis, MN 55414 (US). GOOD, V., Nathaniel, S.; 1011-D 29th Avenue S.E., Minneapolis, MN 55414 (US). SCHAFER, John, B.; 2530 Cleveland St. N.E., Minneapolis, MN 55418 (US). MILLER, Bradley, N.; 13815 58th Avenue N., Plymouth, MN 55446 (US). KONSTAN, Joseph, A.; 582 Cretin Avenue South, St. Paul, MN 55116 (US). HERLOCKER, Jonathan, L.; 939 Old Steine Circle, New Brighton, MN 55112 (US). BORCHERS, Albert, T.; 4912 Zenith Avenue S., Minneapolis, MN 55410 (US). RIEDL, John, T.; 1340 California Avenue West, Falcon Heights, MN 55108 (US).

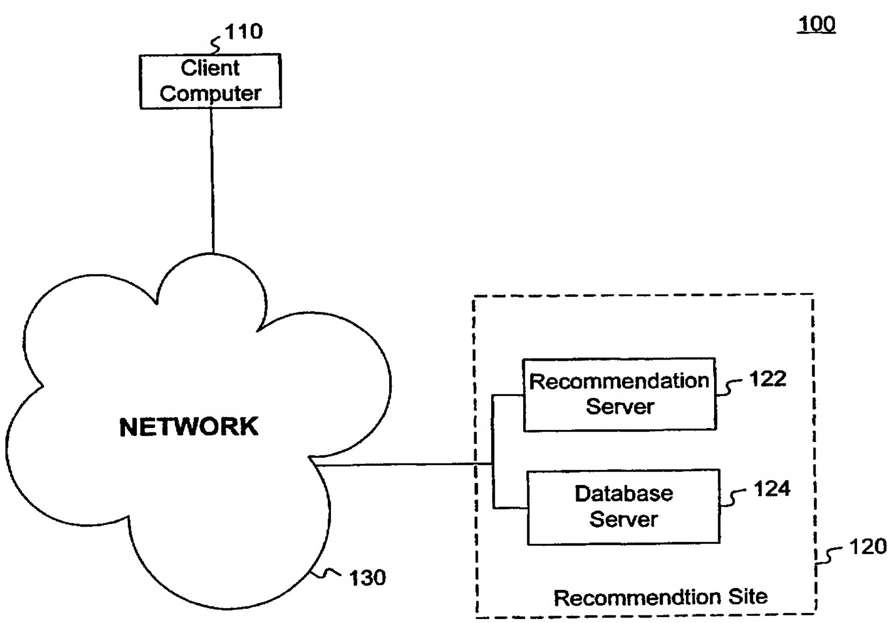
(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, N.W., Washington, DC 20005-3315 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR RECOMMENDING ITEMS TO USERS BASED ON USER PREFERENCES



(57) Abstract: This invention relates to a recommendation system comprising a data processing system (100) connected to a client computer (110), the client computer is connected to a recommendation site (120) via a network (130). A user uses client computer (110) to request and submit information to database server (124) and submit evaluation requests to a recommendation server (122) at recommendation site.

WO 01/37193 A1

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM, METHOD, AND ARTICLE OF MANUFACTURE FOR RECOMMENDING ITEMS TO USERS BASED ON USER PREFERENCES

Background of The Invention

A. Field of the Invention

This invention relates generally to data processing systems and, more particularly, to recommendation systems.

B. Description of the Related Art

Information retrieval (IR) systems allow users to express queries to select documents that match a topic of interest. Some IR systems index a database of documents using the full text of the document or only document abstracts. Sophisticated IR systems rank query results using a variety of heuristics including the relative frequency with which the query terms occur in each document, the adjacency of query terms, and the position of query terms. Other IR systems employ techniques such as term stemming to match words such as "retrieve," "retrieval," and "retrieving." IR systems are generally optimized for ephemeral interest queries, such as looking up a topic in the library. For example, IR systems used on the Internet include AltaVista (www.altavista.com) for web pages and DejaNews (www.deja.com) for discussion list postings. Genetic algorithms have also been used effectively in IR systems and to evolve strategies within a search space as described in Gordon, M., "Probabilistic and Genetic Algorithms in Document Retrieval," Commun. ACM 31, 10.

Information filtering (IF) systems use many of the same techniques as IR systems, but are optimized for long-term information needs from a stream of incoming documents. Accordingly, IF systems build user profiles to describe the documents that should (or should not) be presented to users. Simple examples of IF systems include "kill files" that are used to filter out advertising or flames (i.e., attack messages) and e-mail filtering software that sorts e-mail into priority categories based on the sender, the subject, and whether the message is personal or sent to a list. More complex IF systems provide periodic personalized digests of material from sources such as news wires, discussion lists, and web pages.

Some IF systems use "agents," which are programs that exhibit a degree of autonomous behavior and attempt to act intelligently on behalf of the user for whom they are working. Agents maintain user interest profiles by updating them based on feedback on whether the user likes the items selected by the current profile. For example, *NewT* is a filtering agent for Usenet news based on learning techniques that performs full text analysis of articles using vectorspace technique. More information on *NewT* may be found in Maes P.,"Agents that Reduce Work and Information Overload," CACM, July 1994, hereby incorporated by reference. Another example, *Amalthaea,* is a multi-agent system for personalized filtering, discovery and monitoring of information sources on the Internet. More information on *Amalthaea* may be found in Moukas, A. and Zacharia, G., "Evolving a Multi-Agent Information Filtering Solutions in Amalthaea," Proceedings of Autonomous Agents 97, hereby incorporated by reference. Other examples of feedback generation techniques are probabilistic models, or well-known neural network based learning algorithms.

IR and IF systems can be extremely effective at identifying documents that match a topic of interest, and at finding documents that match particular patterns (e.g., discarding email with the phrase "Make Money Fast" in the title). Unlike human editors, however, these systems cannot distinguish between high-quality and low-quality documents on the same topic. As the number of documents on each topic continues to grow, even the set of relevant documents will become too large to review. For some domains, therefore, the most effective filters must incorporate human judgements of quality.

Recommender systems provide recommendations to users based on various attributes. For example, collaborative filtering (CF) systems are a specific type of recommender system that recommend items to a user based on the opinions of other users. In their purest form, CF systems do not consider the content of an item at all, relying exclusively on the judgement of humans of the item's value. In this way, CF systems attempt to recapture the cross-topic recommendations that are common in communities of people.

Commercial applications of "ratings-based" CF systems now exist in a variety of domains including books, music, grocery products, dry goods, and information. Ratings-

based CF systems are contrasted with text reviews (e.g., reading a review of a movie written by someone else) and with "active" CF systems (e.g., forwarding funny jokes to a set of friends) in that the presence of ratings, either explicitly entered, or implied from behavior allows the CF system to automatically find neighbors for a user. The term neighbors refers to other users who share similar tastes, based on the ratings entered. The identification of neighbors allows the CF system to personalize its recommendations, rather than simply presenting a single overall recommendation, without extra individual effort.

One example of a CF system is the GroupLens Research system that provides CF systems for Usenet news and movies. More information on CF systems may be found at "http://www.grouplens.org."

One of the early computer-based CF systems designed to support a small, close-knit community of users was *Tapestry*. Users could filter all incoming information streams, including e-mail and Usenet news articles. When users evaluated a document, they could annotate it with text, with numeric ratings, and with boolean ratings. Other users could form queries such as "show me the documents that Mary annotated with 'excellent' and Jack annotated with 'Sam should read.'" Another approach is used in Maltz and Ehrlich's *active collaborative filtering* which provides an easy way for users to direct recommendations to their friends and colleagues through a *Lotus Notes* database as described in Maltz, D. and Ehrlich, K., "Pointing the Way: Active Collaborative Filtering," Proceedings of ACM CHI `95.

CF systems for large communities cannot depend on each person knowing all others in a community. Several systems use statistical models to provide personal recommendations of documents by finding a group of other users, known as *neighbors,* that have a history of agreeing with the target user. One example of a statistical model used in CF systems is well-known correlation waited average of normalized ratings described in Herlocker, J., Konstan, J., Borchers, A., Riedl. J., "An Algorithmic Framework for Performing Collaborative Filtering,"Proceedings of the 1999 Conference on Research and Development in Information Retrieval. Once a neighborhood of users is found, particular documents can be evaluated by forming a weighted composite of the neighbors' opinions of that document. Similarly, a user can request recommendations for

a set of documents to read and the system can return a set of documents that is popular within the neighborhood. These statistical approaches, known as automated CF systems, typically rely upon *ratings* as numerical expressions of user preference.

Several ratings-based automated CF systems have been developed. The GroupLens Research system provides an pseudonymous CF solution for Usenet news and movies. *Ringo* and *Video Recommender* are email and web systems that generate recommendations on music and movies respectively. More information about Ringo and Video Recommender may be found at, Shardanand, U., and Maes, P., "Social Information Filtering: Algorithms for Automating `Word of Mouth,'" Proceedings of ACM CHI '95 and Hill, W., Stead, H., Rosenstein, M., and Furnas, G., "Recommending and Evaluating Choices in a Virtual Community of Use," Proceedings of ACM CHI '95, respectively. Indeed, commercial applications of ratings-based collaborative filtering now exist in a variety of domains including books, music, grocery products, dry goods, and information.

Existing recommender systems provide accurate results when the ratings database includes items that have been rated before by users. However, recommender systems provide little or no value when a user is the first one in his neighborhood to enter a rating for an item. Current recommender systems depend on the altruism of a set of users who are willing to rate many items without receiving many recommendations. Moreover, although CF systems are designed to work with a sparse ratings database, areas of unusual sparsity can provide poor results.

Therefore, there exists a need to improve existing recommender systems that contain sparse ratings or unrated items.

Summary of the Invention

Methods, systems, and articles of manufacture consistent with the present invention provide a recommender system that addresses sparsity and early-rater problems by incorporating noncollaborative information filtering techniques into the recommender system. Specifically, a filterbot automated rating agent evaluates new items, and supplements a user rating database by providing ratings for the new items before a user has rated them. To do so, the filterbot agent polls or is notified by various database servers for new items to rate. Once a new item is found, the filterbot agent evaluates the

item based on certain attributes and places the rated item along with an accompanying filterbot ID in a rating database. A recommender system may treat the ratings by the filterbot as if the ratings were provided by a user.

Brief Description of the Drawings

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an implementation of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

Figure 1 depicts a data processing system suitable for practicing methods and systems consistent with the present invention;

Figure 2 depicts a more detailed diagram of the client computer depicted in Fig. 1;

Figure 3A depicts a more detailed diagram of the recommendation server depicted in Fig. 1;

Figure 3B depicts a more detailed diagram of the database server depicted in Fig. 2;

Figure 4 depicts a flow chart of the steps performed by the data processing system of Fig. 1 when providing recommendation in accordance with methods and systems consistent with the present invention;

Figure 5A depicts a more detailed flow chart of the filterbot evaluation order process depicted in Fig. 4;

Figure 5B depicts a more detailed flow chart of the user evaluation process depicted in Fig. 4;

Figure 5C depicts a more detailed flow chart of the learning process depicted in Fig. 4;

Figure 5D depicts a more detailed flow chart of the recommendation process depicted in Fig. 4;

Figure 6 depicts a flow chart of a filterbot model using the filterbot evaluation process of Fig 5A;

Figure 7 depicts a flow chart of a second filterbot model using the filterbot evaluation process of Fig 5A;

Figure 8 depicts a flow chart of a third filterbot model using the learning process of Fig 5C; and

Figure 9 depicts a flow chart of a fourth filterbot model using the learning process of Fig 5C.

Detailed Description of the Preferred Embodiment

The following detailed description of the invention refers to the accompanying drawings. Although the description includes exemplary implementations, other implementations are possible, and changes may be made to the implementations described without departing from the spirit and scope of the invention. The following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims. Wherever possible, the same reference numbers will be used throughout the drawings and the following description to refer to the same or like parts.

Overview

Methods and systems consistent with the present invention address the rating sparsity and early rater problems by incorporating non-collaborative IF techniques into a recommender system, such as a CF system. IF techniques are introduced through the creation of "filterbots." A filterbot is an automated rating robot that evaluates and rates new items. Filterbots, and CF systems in general, may use different scales to reflect different types and levels of information available for discerning among items, such as a unary scale, binary scale, or a Likert scale

A unary scale is often used in cases where positive information is available for some items, but other items have no information available. For example, purchase records are often converted into unary ratings. Items that are bought are rated positively; for all other items, no information is available. A binary scale is often used in cases where items can be classified into "good" and "bad" (and optionally unknown), but not to different degrees of good and bad. For example, if a user has expressed interest in comedies, any movie that is a comedy would be rated 1 (good) and any movie that is not would be rated 0 (bad). Likert scale ratings (scales such as 1 to 5 or 1 to 7) allow a wider range of ratings. For example, if movies are to be rated based on box office success, then the top blockbusters could be rated 5, successful movies rated 4, average movies rated

3, etc. Different algorithms exist in the recommender system to perform neighbor selection and prediction in different rating scales.

A filterbot is termed *reflective* if it subsequently changes ratings that it has assigned to items and *non-reflective* if ratings are not revised. For example, a binary filterbot that rates news articles based on whether the term "Kosovo" appears within the article is non-reflective. On the other hand, a Likert filterbot that groups movies into five equal-sized categories based on box-office sales is reflective since it re-rates movies as other movies adjust the cut-offs. Non-reflective filterbots more closely model the ratings behavior of human users, and may exhibit greater correlation stability. Other dimensions in which filterbots can be categorized include the frequency of update (as items are added, periodic, one-time); whether they are customized to match a particular demographic segment or individual; and whether they use machine learning vs. static evaluation algorithms.

A filterbots may be created by various people to meet various goals. For example, web system administrators may create filterbots to enhance their e-commerce applications, or an end user may create a filterbot to help personalize.

Filterbots may also be based on feedback generation techniques, such as probabilistic models, well-known neural network based learning algorithms, or statistical models, described above. Filterbots may also be based on machine learning technology, well-known rule-induction learning, and data mining techniques, described below. Genetic algorithms may also be used to produce a filterbot rating. More information on genetic algorithms can be found in Forrest, S., "Genetic Algorithms," ACM Comput. Surv. 28, 1.

System Components

Fig. 1 depicts a data processing system 100 suitable for practicing methods and systems consistent with the present invention. Data processing system 100 comprises a client computer 110 connected to a recommendation site 120 via a network 130, such as the Internet. The user uses client computer 110 to request and submit information to database server 124 and submit evaluation requests to a recommendation server 122 at recommendation site 120.

Although only one client computer 110 is depicted, one skilled in the art will appreciate that data processing system 100 may contain many more client computers. One skilled in the art will also appreciate that client computer 110 may come with recommendation server software already installed.

Figure 2 depicts a more detailed diagram of client computer 110, which contains a memory 220, a secondary storage device 230, a central processing unit (CPU) 240, an input device 250, and a video display 260. Memory 220 includes browser 222 that allows users to interact with recommendation server 122 and database server 124 by transmitting and receiving files, such as web pages. A web page may include images or textual information to provide an interface to receive ratings and requests for evaluations from a user using hypertext markup language (HTML), Java or other techniques. An example of a browser suitable for use with methods and systems consistent with the present invention is the Netscape Navigator browser, from Netscape.

As shown in Figure 3A, recommendation server 122 includes a memory 310, a secondary storage device 316, a CPU 326, an input device 328, and a video display 330. Memory 310 includes recommendation engine 312, which determines if an item should be recommended to the user. Recommendation engine 312 may use many different techniques to generate recommendations based on user interest profiles. One technique that may be used to generate recommendations is automated collaborative filtering as described in Resnick, Iacovo, Susha, Bergstrom, and Riedl, "GroupLens: An Open Architecture For Collaborative Filtering Of Netnews," Proceedings of the 1994 Computer Supported Collaborative Work Conference (1994). Other recommendation techniques are described in U.S. application serial no. 08/729,787, filed October 8, 1996, U.S. application serial no. 08/733,806, filed October 18, 1996, attorney docket no. 7744-6000, filed September 23, 1999, attorney docket no. 7744-0009, filed September 24, 1999, attorney docket no. 7744-0006, filed November 12, 1999, all incorporated by reference. Recommender systems may also be based on well-known CF systems, logical rules derived from data, or on statistical or machine learning technology. For example, a recommender system may use well-known rule-induction learning, such as Cohen's Ripper, to learn a set of rules from a collection of data as described in Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., "Combining

Collaborative Filtering with Personal Agents for Better Recommendations," Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99). Recommender systems may also be based on well-known data mining techniques that include a variety of supervised and unsupervised learning strategies and produce "surprising" results expressed as assoications or rules embedded in a data set. Recommender systems may also contain rating functions (models) programmed by a system administrator. The rating functions are either a formula or a table of ratings that determines business goals (e.g., the formula may specify a low rating for low-stock and out-of-stock items). These mentioned systems also require user data as input to produce personalized recommendations for users.

Recommendation engine 312 also receives evaluations from filterbot engine 314 and client computer 110. To receive the evaluations, recommendation engine 312 may use a web page, Application Program Interfaces (API), or other input interface. An API is a set of routines, protocols, or tools for communicating with software applications. APIs provide efficient access to the recommendation engine without the need for additional software to interface with the recommendation engine. Evaluations may come in various forms. For example, an evaluation may be a rating on a unary scale. Also, an evaluation may be based on user purchase data. That is, the evaluation would include a list of items recently purchased by the user.

Also contained in memory 310 is a filterbot engine 314, which monitors database server 124 for new items and evaluates them. Filterbot engine 314 receives the new items and evaluates them using a filterbot model 324 stored in database 318. A filterbot model is a preprogrammed evaluation algorithm based on attributes of items. When performing an evaluation, filerbot model 324 may also include external attributes, such as other user ratings. One skilled in the art will appreciate that filterbot engine 314 may supply a rating to recommendation engine 312 by using various APIs. One skilled in the art will also appreciate that recommendation engine 312 may include filterbot engine 314 to provide on-demand evaluations for new items when needed by recommendation engine 312. Secondary storage device 316 includes a database 318 that stores user ratings in user rating file 320 filterbot ratings in a filterbot rating file 322, and a filterbot model 324. One skilled in the art will appreciate that filterbot model 324 may be

represented in any manner that can be used to rate items based on characteristics, including lists of characteristic values, characteristic value weights, neural networks that receive item characteristics to produce a rating, and other representations.

As shown in Figure 3B, database server 124 includes a memory 332, a secondary storage device 336, a CPU 342, an input device 344, and a video display 346. Memory 332 includes database software 334 that provides access to database 338 in secondary storage device 336. An example of such a program suitable for use with methods and systems consistent with the present invention is the Sybase Adaptive Server Enterprise from Sybase, of Emeryville, California. Database 338 includes items table 340, which holds both rated and unrated items. For example, items table 340 may contain the entire list of Usenet documents, or an online bookstore's catalog of books.

Although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects may be stored on or read from other computerreadable media, such as secondary storage devices, like hard disks, floppy disks, and CD-ROM; a carrier wave received from a network like the Internet; or other forms of ROM or RAM. Additionally, although specific components and programs of client computer 110, recommendation server 122, and database server 124 have been described, one skilled in the art will appreciate that these may contain additional or different components or programs.

Overview of the Recommendation Process

Figure 4 depicts a flow chart of the steps performed by recommendation site 120. The recommendation process is initiated by a filterbot evaluation process (step 402). The filterbot evaluation process comprises receiving items to evaluate from a database and the evaluation of these items. The filterbot evaluation process is completed by adding a rating to a filterbot rating database. Next a user evaluation process is started (step 404). This process entails various users viewing items from a database, providing an evaluation for the item, and placing the evaluation for the item in a user rating database. Essentially, filterbot evaluation process 402 and user evaluation process 404 may occur simultaneously, each rating items from the database. Since user ratings are generally preferred over filterbot ratings, if a user has provided a rating for an item, a learning process may update the filterbot model to incorporate the user evaluations to apply to

future evaluations, and rerate all items previously rated by the filterbot engine using the updated filterbot model (step 406). Finally, a recommendation process may receive a request for a recommendation from a user, and provide a recommendation based on the user's preferences, the preferences of other users, or the ratings provided by the filterbot (step 408). Although the modified recommendation process is shown in a particular order, one skilled in the art will appreciate that any order for steps 402, 404, 408, and 408 may occur.

Further details and operations of the modified recommendation process will now be explained with reference to the flowcharts of Figures 5A-5D.

Filterbot Evaluation Process

As shown in Fig. 5A, filterbot evaluation process 502 is initiated, for example, by filterbot engine 314 obtaining a new item from database server 124 (step 502). To do so, filterbot engine 314 may communicate with database server 124 through an API. Database server 124 may provide only new items to filterbot engine 314 by using well-known detection mechanisms, such as Usenet range files. One skilled in the art will appreciate that filterbot engine 314 may communicate and retrieve items from database 124 by other means, such as the well-known HTTP interface.

Once filterbot engine 314 obtains a new item to evaluate, filterbot engine 314 applies a filterbot model 324 to the new item (step 504). The filterbot model determines whether the new item contains certain characteristics (step 506), and if so, rates the item a "1" (step 508). Otherwise, the item is rated a "0" (step 510). After the item has been assigned a rating, filterbot engine 314 supplies the rating and a corresponding identification number to recommendation engine 312 (step 512). The rating/identification pair are stored in filterbot rating file 322. The rating/identification pair are stored in an identical manner as a user submitting an evaluation, further described below.

One embodiment of a filterbot model 324 is shown in Fig. 6. In this embodiment, filterbot engine 314 is preprogrammed to rate a set of authors highly. As a new document arrives (step 602), filterbot model 324 checks to see whether that author of the document appears in the "liked_author" list (step 604). If so, the model rates the document as "1" (step 606). If no author in the "liked_author" matches the author of the

new document, filterbot model 324 rates the document as "0" (step 608). Finally, filterbot model 324 returns the rating to recommendation engine 312 (step 610).

Another embodiment of a filterbot model 324 is shown in Fig 7. In this embodiment, the model initializes "this_doc" to zero (step 702). The "this_doc" list stores the frequency of occurrences of all words in the document. Once initialized, the filterbot model may accept a new document and place the contents of the document in "docs"(step 704). Filterbot model 324 removes all stopwords from the document (step 706). A stopword is a word that will be excluded from the filtering process. For example, a stopword may be any article, such as "the" or "a." Next, filterbot model 324 tallies the occurrences of all remaining words in the document, keeping the frequency of words in "this_docs" and incrementing the total count in "all_docs" (step 708). Once all words are examined and the "this_doc" and "all_docs" lists are completed, for each word in the "goal_terms" list (step 708), filterbot model 324 calculates the percentage of goal term occurrences that occur within this document (step 710). The "goal_terms" are the prespecified words that filterbot model 324 uses to identify documents of interests. Finally, filterbot model 324 computes a document score "doc_score" which reflects the degree to which the goal terms are concentrated in that document. The score is normalized (e.g., mapped to the appropriate rating scale) and filterbot model 324 supplies the results to recommendation engine 312 as a rating (step 712).

User Evaluation Process

As shown in Fig. 5B, user evaluation process 404 is initiated by displaying a list of items on browser 224 (step 514). For example, client computer 110 may use browser 224 to communicate with database software 334 to retrieve a list of items for the user. To do so, client computer 110 may use well-known document server APIs, such as Network News Transfer Protocol (NNTP), or HTTP. Once a list of items are displayed on browser 224, the user may evaluate some, or all of the items presented (step 516). The user explicitly assess the value of the item by providing a rating for each item. Alternatively, a rating may be implied by recording various parameters, such as "click-throughs," time spent at a particular web page, or shopping cart contents. A shopping cart allows a user to select items and purchase the items in a well-known web interface. Database 338 may also contain various "click stream" data about a user to send to

recommendation engine 312 when the user requests a recommendation.

Regardless of the method used to evaluation an item, the results are submitted to recommendation engine 312 (step 518). The results include a rating and a user identification, and are stored in user rating file 320. Similar to step 514, client computer 110 and database server 124 may communicate with recommendation engine 312 though an API. The API is provided to enter ratings for particular items into user rating file 320.

Filterbot Learning Process

As shown in Fig. 5C, each time a user evaluates an item, learning process 406 may be initiated to update filterbot model 324 and rerate items in filterbot rating file 322. By updating filterbot model 324, the model may more accurately rate items in a manner that matches the model. Each time a user rates an item, learning process 406 runs. Learning process 406 first checks filterbot rating file 322 to determine whether filterbot engine 314 has already rated the item (step 520). If filterbot engine 314 has not yet rated the item, there is no need to update filterbot model 324. However, if filterbot engine 314 previously rated the item, the item's characteristics are added to filterbot model 324.

Once filterbot model 324 is updated to include the user's ratings and characteristics (step 522), learning process 406 may rerate items in filterbot rating file 322 to increase the accuracy of the model's adherence to the specification (step 524). Since filterbot model 324 has been updated, presumably the item would be rated differently. However, to conserve processing time, the updated filterbot model 324 may continue rating using the new information. Finally, if filterbot engine 314 is to update the filterbot rating file 322, evaluation process 402 may be re-run for all items in the rating file (step 526).

One embodiment of filterbot model 324 that may be updated in shown in Fig. 8. In this embodiment, filterbot engine 314 takes a set of unary ratings, such as liked documents, and builds a filterbot model 324 that rates highly documents written by any author of a liked document. The learning process starts by accepting a new rating for a document from a user (step 802). The learning process determines if the author of the rated document is listed in the "liked_author" list. (step 804). If the author is listed in the "liked_author" list then filterbot model 324 is current and the learning process is completed (step 806). However, if the author is not listed in the "liked_author" list, the

author is added to the list (step 808). Once the author is added to the "liked_author" list, filterbot engine 314 may rerate items in filterbot rating file 324 (step 810).

Fig. 9 depicts a second embodiment to update filterbot model 324. In this embodiment, filterbot model 324 may be updated when a user submits a rating for a document (step 902), or when a new document is submitted (step 912). When a new rating is submitted for a document, filterbot model 324 places the rated document in "rdocs" (step 904). Filterbot model 324 then removes all stopwords from the document (step 906). Next, filterbot model 324 tallies the occurrences of all remaining words in the document, keeping the frequency of words in "rated_docs" and "rated_wordcount"(step 908). Once all words in the document have been tallied, filterbot model 324 determines if the document has been previously evaluated by filterbot engine 314 (step 910). If so, filterbot model 324 may accept the document for evaluation (step 912), and place the contents of the document in "docs"(step 914). Otherwise, filterbot engine 314 has already rated the document, and filterbot model 324 may begin updating filterbot rating file 322 (step 920).

Similar to step 704, if the document has not yet been rated, filterbot model 324 removes all stopwords from the document (step 916). Next, filterbot model 324 tallies the occurrences of all remaining words in the document, keeping the frequency of words in "this_docs" and incrementing the total count in "all_docs" (step 918). Once all words are examined and the "this_doc" and "all_docs" lists are completed, filterbot model 324 may update filterbot rating file 322 (step 920).

The "regen" process 920 is a learning model in which the model learns a set of keywords that best reflect the characteristics of documents that have been rated (in this case, unary ratings --any rated document is good). In step 922, "temp_words" list is equal to the set of words in the "rated_docs" list. filterbot model 324 then sorts the set of words that occur in the rated document by a score that reflects their selectivity, using the formula in step 924.

For example, if there are 10000 words in the rated documents and "Kosovo" appears 100 times; and if there are 1000000 words in all documents and "Kosovo" appears 1000 times, then the ratio is (100/10000)/(1000/1000000) which is .01 / .001 which is10. In other words, Kosovo appears 10 times as often in rated documents than

in unrated ones. Regen process 920 sorts by that score, and returns up to "max_keywords" of them, as long as they are above the "min_ratio" -- For example, if min_ratio is 2 and max_keywords is 10, then it returns up to the 10 most selective words, as long as each word is at least twice as common in selected documents than unselected (step 926).

One skilled in the art will appreciate that the filterbot model in Fig. 9 may also remove ratings from filterbot rating database 324.

Recommendation Process

As shown in Fig. 5D, once filterbot engine 314 provides rated items to filterbot rating file 322, and the user provides rated items to user rating file 320, recommendation engine 312 may begin providing recommendations. The first step is to obtain ratings information from user requesting a recommendation from user rating file 320 (step 528). If no ratings are available for the user, processing ends and a default list is supplied as a recommendation. A default list may be a predesignated list of items. Recommendation engine 312 uses the data from user rating file 320 and filterbot rating file 322 to locate potential neighbors (step 530). The term "neighbor" means another user in user rating file 320, or another filterbot entry in filterbot rating file 322 with similar interests as the user. For example, if a filterbot model 324 has rated similar items as the user, that entity may be considered a potential neighbor. Alternatively, recommendation engine 312 may locate potential neighbors in only the filterbot rating file or only the user rating file. One skilled in the art will appreciate that the user rating file and the filterbot rating file may be a combined, or a separate file. At this point, an entity is considered a potential neighbor since the affinity between the user and the entity still needs to be determined, as further described below. For example, the ideal neighbor for a user would have rated many items that the user has also rated and rated them similarly. If no potential neighbors are found (step 532), recommendation engine 312 randomly picks a predetermined number of entities in interest data table 324 to substitute as potential neighbors (step 534). The substituted neighbors are randomly selected and used to provide recommendations.

If, however, a potential neighbor is found (step 532), recommendation engine 312 computes an affinity between the user and the potential neighbor using an appropriate

affinity algorithm (step 536). The affinity algorithms provides an affinity value that indicates how similar the user and entity are in terms of preferences. One skilled in the art will appreciate that any well-known affinity algorithm used in standard recommender systems may be used to compute an affinity, such as correlation on normalized ratings.

After each affinity value is computed for a user and a potential neighbor, recommendation engine 312 determines if the affinity value is above a predetermined threshold value (step 538). One skilled in the art will appreciate that the threshold value may be a maximum value, minimum value, or a range of values. If the affinity value is above the threshold value, the potential neighbor is added to a neighbor list (step 540). Each neighbor on the neighbor list is used to provide rating information to compute a recommendation for the user. Otherwise, if the affinity value is below the threshold value, the potential neighbor is dropped and the next potential neighbor is located in user rating file 320 and filterbot rating file 322 (step 530).

Recommendation engine 312 located neighbors until enough neighbors have been located (step 542). For example, to provide a quick recommendation, recommendation engine 312 may require ten neighbors. However, to provide a more accurate recommendation, recommendation engine 312 may require fifty neighbors. Once the requisite number of neighbors has been located, recommendation engine 312 may provide a recommendation to the user using well-known recommendation techniques (step 544).

By integrating filterbot engine ratings into recommender systems, the utility of recommendations will increase to other users who agree with the filterbot's selections.

Conclusion

Methods, systems, and articles of manufacture consistent with the present invention provide a recommender system that addresses sparsity and early-rater problems by incorporating noncollaborative information filtering techniques into the recommender system. Specifically, a filterbot automated rating agent evaluates new items, and supplements a user rating database by providing ratings for the new items before a user has rated them. To do so, the filterbot agent polls various database servers for new items to rate. Alternatively, the agent is notified by the database of a new document. Once a new item is found, the filterbot agent rates the item based on certain attributes and places

the rated item along with an accompanying filterbot ID in a rating database. A recommender system may treat the rated item as if the item were evaluated by a user.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone.

Claims

1. A system for evaluating items based on user preferences and item characteristics, comprising:

a filterbot subsystem comprising:

evaluation means for evaluating items;

producing ratings means for producing ratings of items based on characteristics associated with the items;

a recommendation subsystem comprising:

a first interface means for receiving user preference data;

a second interface means for receiving requests for item evaluations;

processing means for processing the received requests for item evaluations; and

a third interface means for presenting the item evaluations to a user;

2. The system of claim 1, wherein the processing means computes evaluations based on the user's preferences and other users' preferences.

3. The system of claim 1, wherein the processing means computes evaluations based on the user's preferences and the ratings of items.

4. The system of claim 1, wherein the user preferences are expressed as Likert scale ratings, unary ratings, or binary ratings.

5. The system of claim 1, wherein the filterbot ratings are expressed as Likert scale ratins, unary ratings, or binary ratings.

6. The system of claim 1, wherein the filterbot subsystem further contains:

interface means for receiving preference data.

7. The system of claim 1, wherein the filterbot subsystem and recommendation subsystem are integrated.

8. The system of claim 1, wherein the producing ratings means rates an item only once.

9. The system of claim 1, further comprising:

updating means that updates the filterbot model, and wherein the producing ratings means may rerate items when the filterbot model has been updated.

10. The system of claim 1, wherein the filterbot subsystem further contains:

learning means that revise the filterbot model according to user preference data received from the recommendation engine.

11. The system of claim 1, wherein the producing ratings means produces ratings based on item characteristics, and a factor selected from the group consisting of: a model, user preference data received from the recommendation engine, and popularity data.

12. The system of claim 1, wherein the evaluation means evaluates with an abstract model.

13. The system of claim 12, wherein the abstract model determines the preferences of at least one user with a neural network.

14. The system of claim 12, wherein the abstract model determines the preference of at least one user with genetic algorithms.

15. The system of claim 12, wherein the abstract model determines the preference of at least one user with a statistical model.

16. The system of claim 12, wherein the abstract model determines the preference of at least one user with a learning model.

17. The system of claim 1, wherein the evaluation means contains user-programmed rating functions programmed by an end-user.

18. The system of claim 1, wherein the evaluation means contains user-programmed rating functions programmed by a system administrator.

19. The system of claim 1, wherein the evaluation means contains rules derived from data mining techniques.

20. The system of claim 1, wherein the recommendation subsystem contains a database that includes a plurality of filterbot ratings and a plurality of user ratings.

21. The system of claim 1, wherein the recommendation subsystem contains a first database that includes a plurality of filterbot ratings and a second database that includes plurality of user ratings.

22. The system of claim 1, wherein the user preference data relates to one of web pages, books, click-through data, or purchase data.

23. A method for providing a recommendation for a plurality of items for a user based on user preferences and item characteristics executed in a data processing system.

comprising the steps of:

evaluating an item to obtain a corresponding rating based on characteristics associated with the item;

obtaining a recommendation request;

generating a recommendation in response to the request based on at least the user preferences; and

providing the recommendation to the user.

24. The method of claim 23, wherein generating a request further includes:

generating a recommendation based on user preferences of other users.

25. The method of claim 23, wherein generating a request further includes:

generating a recommendation based on the evaluated item.

26. The method of claim 23, wherein the user preferences are expressed as Likert scale ratings, unary ratings, or binary ratings.

27. The method of claim 23, wherein the corresponding ratings are expressed as Likert scale ratings, unary ratings, or binary ratings.

28. The method of claim 23, wherein evaluating an item further includes:

receiving item to be rated from a first interface, wherein obtaining a recommendation request further includes receiving the request from a second interface.

29. The method of claim 27, wherein the first interface and the second interface are the same interface.

30. The method of claim 23, wherein evaluating an item further includes:

rating the item only once.

31. The method of claim 23, further containing:

updating the filterbot model, and wherein evaluating an item further includes rerating an item when the filterbot model has been updated.

32. The method of claim 23, further containing:

rerating items with the filterbot model according to user preference data received from the recommendation engine.

33. The method of claim 23, wherein evaluating an item further includes:

producing a rating based on item characteristics, a model, or user preference data received from the recommendation engine, and popularity data.

34. The method of claim 23, wherein evaluating an item further includes:
evaluating an item with an abstract model.

35. The method of claim 34, wherein the abstract model includes a neural network, and
wherein filtering an item further includes determining the preference of at least one user with the abstract model.

36. The method of claim 34, wherein the abstract model includes genetic algorithms, and
wherein filtering an item further includes determining the preference of at least one user with the abstract model.

37. The method of claim 34, wherein the abstract model includes a statistical model, and
wherein filtering an item further includes determining the preference of at least one user with the abstract model.

38. The method of claim 34, wherein the abstract model includes a rule-induction learning model, and wherein filtering an item further includes determining the preference of at least one user with the abstract model.

39. The method of claim 23, wherein evaluating an item further includes:
evaluating an item using user-programmed rating functions programmed by an end-user.

40. The method of claim 23, wherein evaluating an item further includes:
evaluating an item using user-programmed rating functions programmed by a system administrator.

41. The method of claim 23, wherein evaluating an item further includes:
evaluating an item using rules derived from data mining techniques.

42. The method of claim 23, wherein evaluating an item further includes:
submitting the rating to a database that includes a plurality of filterbot ratings and a plurality of user ratings.

43. The method of claim 23, wherein evaluating an item further includes:
submitting the rating to a database that includes a plurality of filterbot ratings.

44. The method of claim 23, wherein generating a recommendation further includes:

using user preference of the requesting user and user preference of another user.

45. The method of claim 23, wherein generating a recommendation further includes:

using user preference of the requesting user ratings and the rating for the item.

46. The method of claim 23, wherein the user preference data relates to one of web pages, books, click-through data, or purchase data.

47. A computer readable medium for controlling a data processing system to perform a method for providing a recommendation for a plurality of items for a user based on user preferences and item characteristics executed in a data processing system, the computer readable medium comprising:

an evaluation module for evaluating an item to obtain a corresponding rating based on characteristics associated with the item;

an obtaining module for obtaining a recommendation request;

a generating module for generating a recommendation in response to the request based on at least the user preferences; and

a providing module for providing the recommendation to the user.

48. The computer readable medium of claim 47, wherein the generating module further includes generating a recommendation based on user preferences of other users.

49. The computer readable medium of claim 47, wherein the generating module further includes generating a recommendation based on the evaluated item.

50. The computer readable medium of claim 47, wherein the user preferences are expressed as Likert scale ratings, unary ratings, or binary ratings.

51. The computer readable medium of claim 47, wherein the corresponding ratings are expressed as Likert scale ratings, unary ratings, or binary ratings.

52. The computer readable medium of claim 47, wherein the evaluating module further includes receiving item to be rated from a first interface, and wherein the obtaining module further includes receiving the request from a second interface.

53. The computer readable medium of claim 50, wherein the first interface and the second interface are the same interface.

54. The computer readable medium of claim 47, wherein the evaluating module further includes rating the item only once.

55. The computer readable medium of claim 47, further including:

an updating module for updating the filterbot model, and wherein the producing ratings means may rerate items when the filterbot model has been updated.

56. The computer readable medium of claim 47, further comprising:

a learning module for revising the filterbot model according to user preference data received from the recommendation engine.

57. The computer readable medium of claim 47, wherein the evaluating module further includes:

producing a rating based on item characteristics, a model, or user preference data received from the recommendation engine, and popularity data.

58. The computer readable medium of claim 47, wherein the evaluating module further includes:

evaluating an item with an abstract model.

59. The computer readable medium of claim 56, wherein the abstract model includes a neural network, and wherein the evaluating module further includes determining the preference of at least one user with the abstract model.

60. The computer readable medium of claim 56, wherein the abstract model includes genetic algorithms, and wherein the evaluating module further includes determining the preference of at least one user with the abstract model.

61. The computer readable medium of claim 56, wherein the abstract model includes a statistical model, and wherein the evaluating module further includes determining the preference of at least one user with the abstract model.

62. The computer readable medium of claim 56, wherein the abstract model includes a rule induction learning model, and wherein the evaluating module further includes determining the preference of at least one user with the abstract model.

63. The computer readable medium of claim 47, wherein the evaluating module further includes:

evaluating an item using user-programmed rating functions programmed by an end-user.

64. The computer readable medium of claim 47, wherein the evaluating module further includes:

evaluating an item using user-programmed rating functions programmed by a system administrator.

65. The computer readable medium of claim 47, wherein the evaluating module further includes:

evaluating an item using rules derived from data mining techniques.

66. The computer readable medium of claim 47, wherein the evaluating module further includes:

submitting the rating to a database that includes a plurality of filterbot ratings and a plurality of user ratings.

67. The computer readable medium of claim 47, wherein the evaluating module further includes:

submitting the rating to a database that includes a plurality of filterbot ratings.

68. The computer readable medium of claim 47, wherein the generating module further includes:

using user preference of the requesting user and user preference of another user.

69. The computer readable medium of claim 47, wherein the generating module further includes:

using user preference of the requesting user ratings and the rating for the item.

70. The computer readable medium of claim 47, wherein the user preference data relates to one of web pages, books, click-through data, or purchase data.

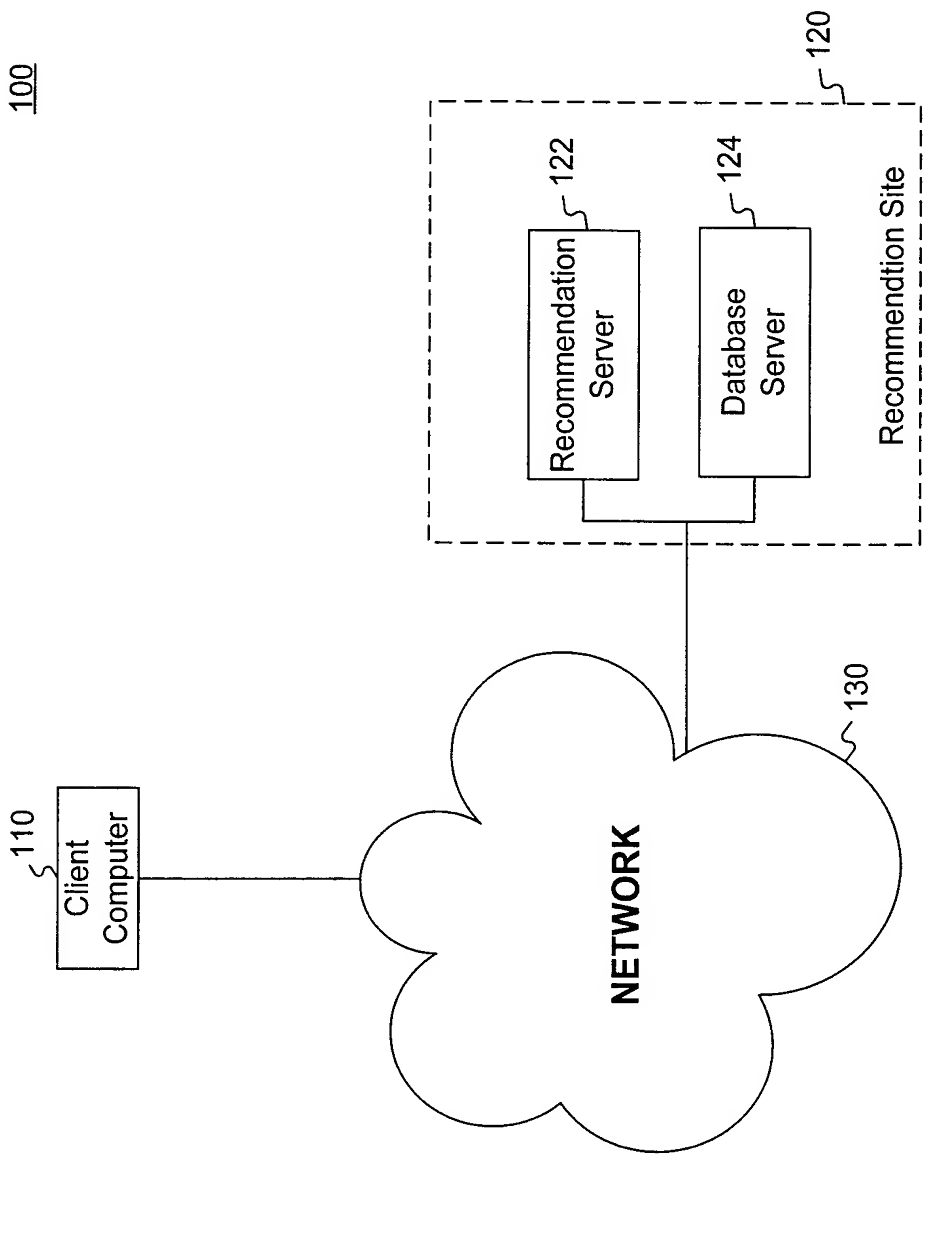

100
120
122
Recommendation Server
124
Database Server
Recommendtion Site
110
Client Computer
NETWORK
130

FIG. 1

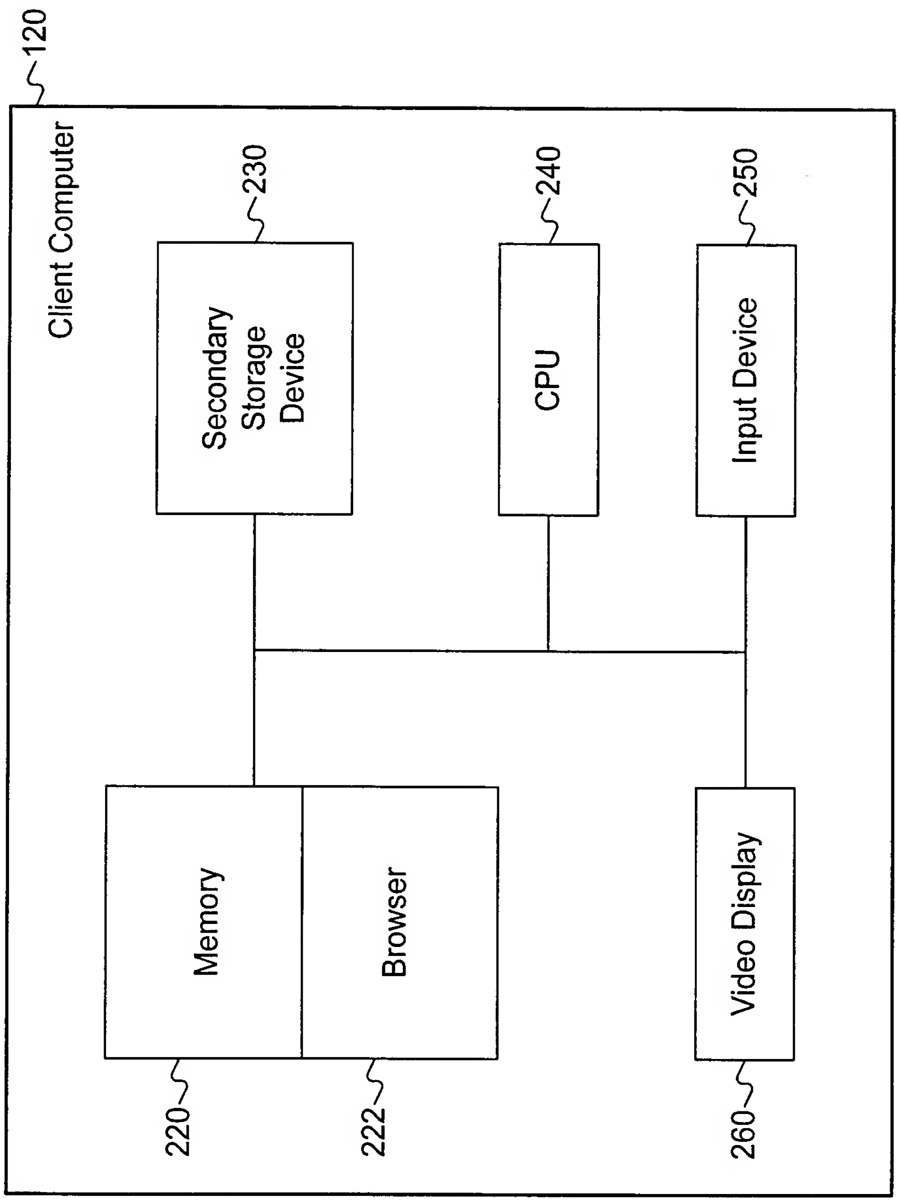

120
Client Computer
Secondary Storage Device
230
CPU
240
Input Device
250
Memory
220
Browser
222
Video Display
260

FIG. 2

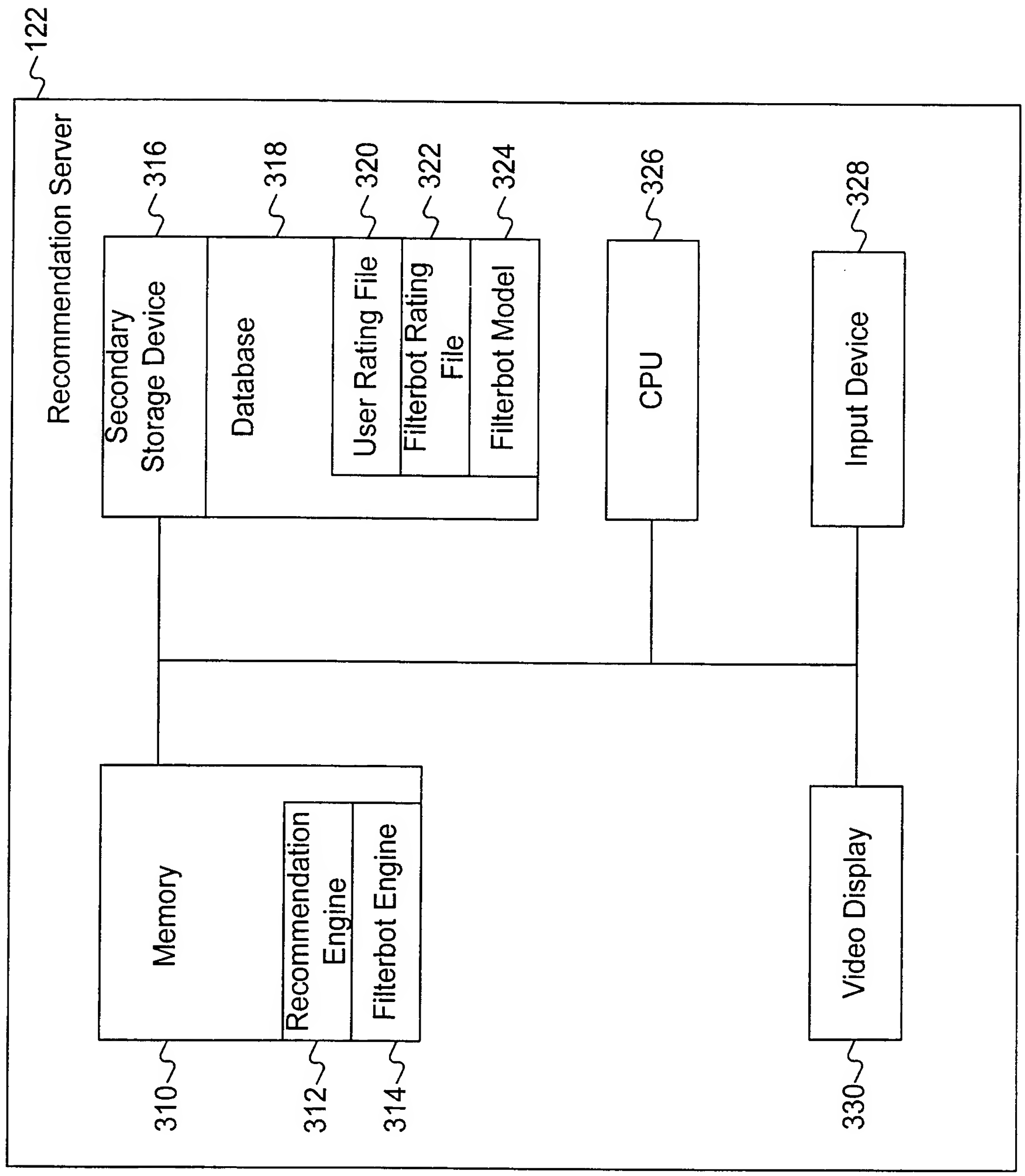

122
Recommendation Server
Secondary Storage Device
316
Database
318
User Rating File
320
Filterbot Rating File
322
Filterbot Model
324
CPU
326
Input Device
328
Memory
310
Recommendation Engine
312
Filterbot Engine
314
Video Display
330

FIG. 3A

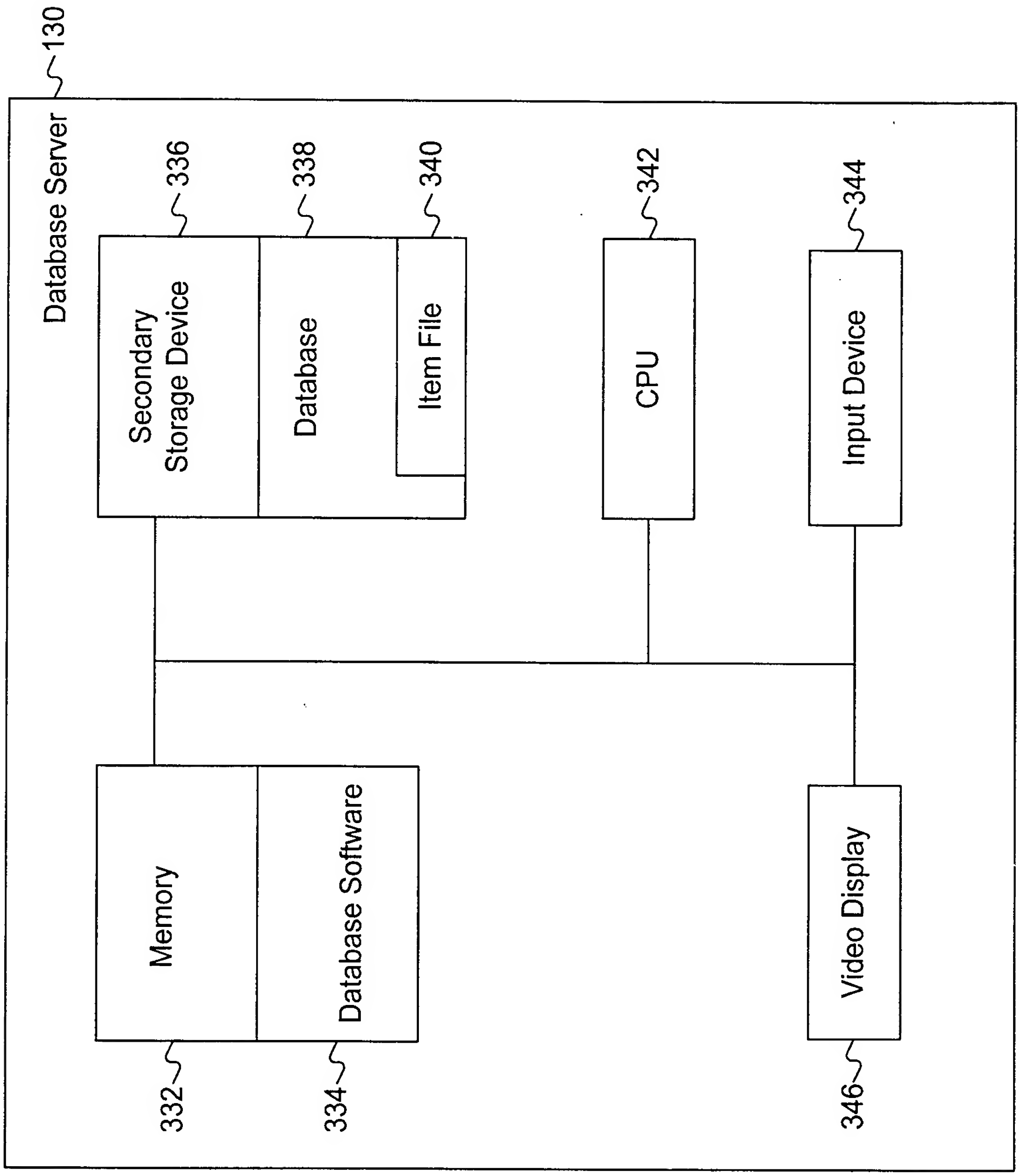

130
Database Server
Secondary Storage Device
336
Database
338
Item File
340
CPU
342
Input Device
344
Memory
332
Database Software
334
Video Display
346

FIG. 3B

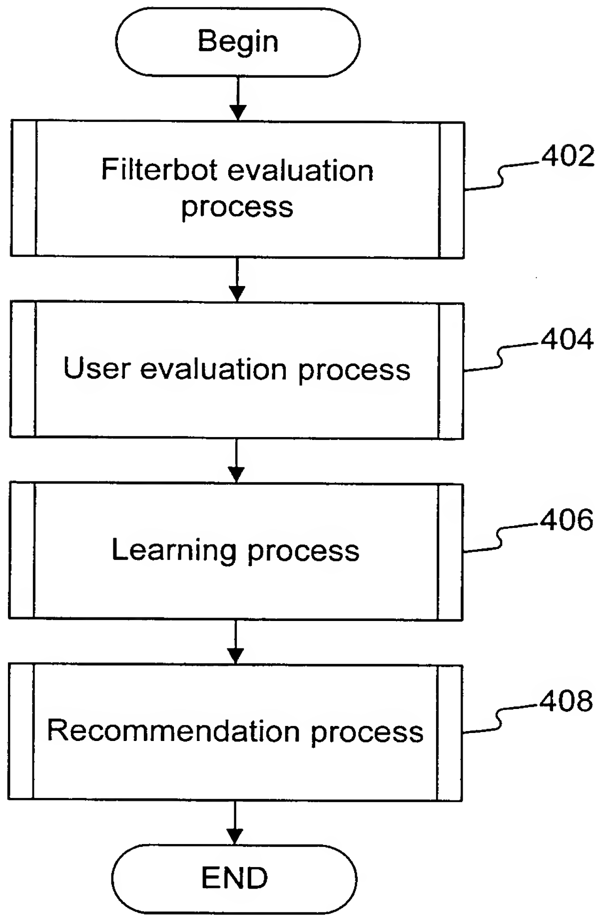

Begin
Filterbot evaluation process
402
User evaluation process
404
Learning process
406
Recommendation process
408
END

Fig. 4

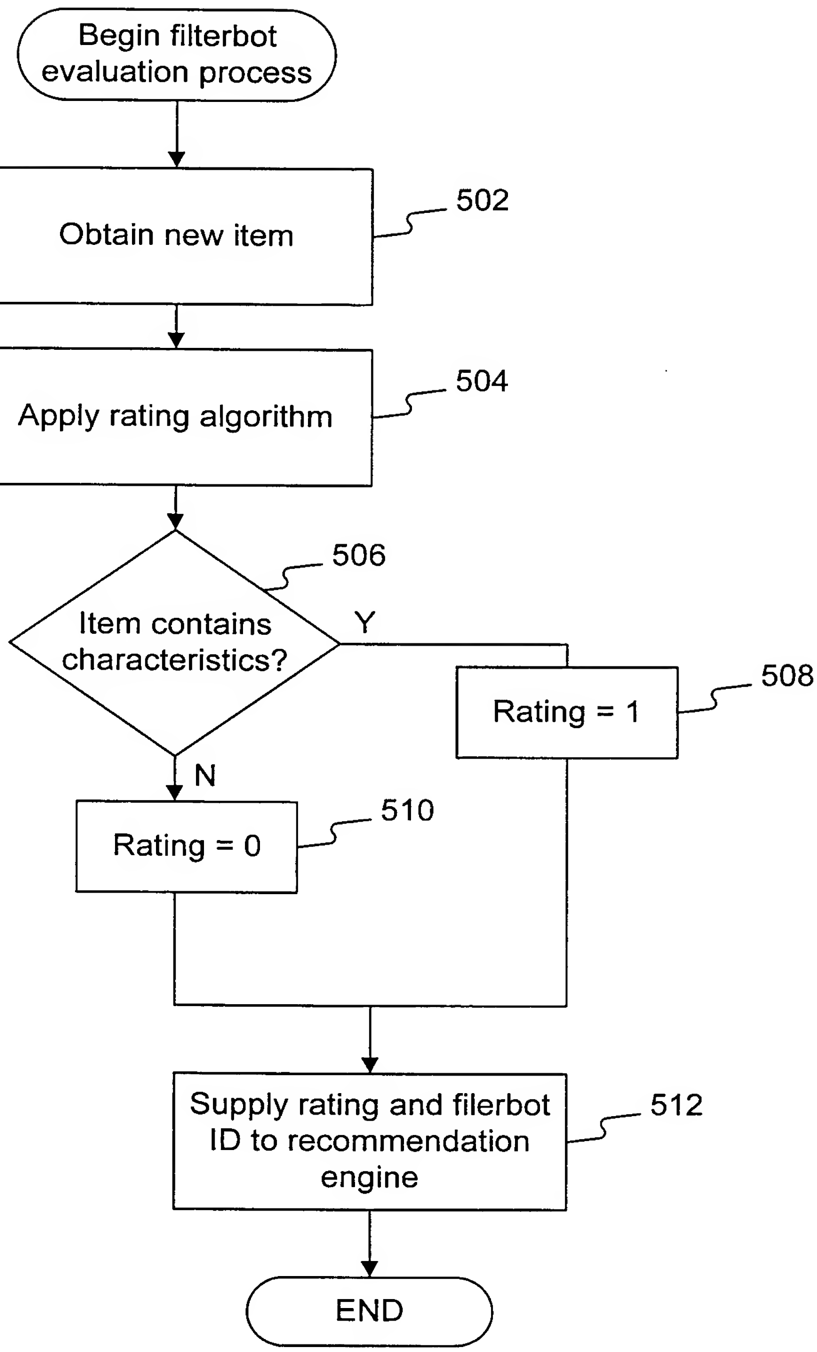

Begin filterbot evaluation process
Obtain new item
502
Apply rating algorithm
504
Item contains characteristics?
506
Y
N
Rating = 1
508
Rating = 0
510
Supply rating and filerbot ID to recommendation engine
512
END

Fig. 5A

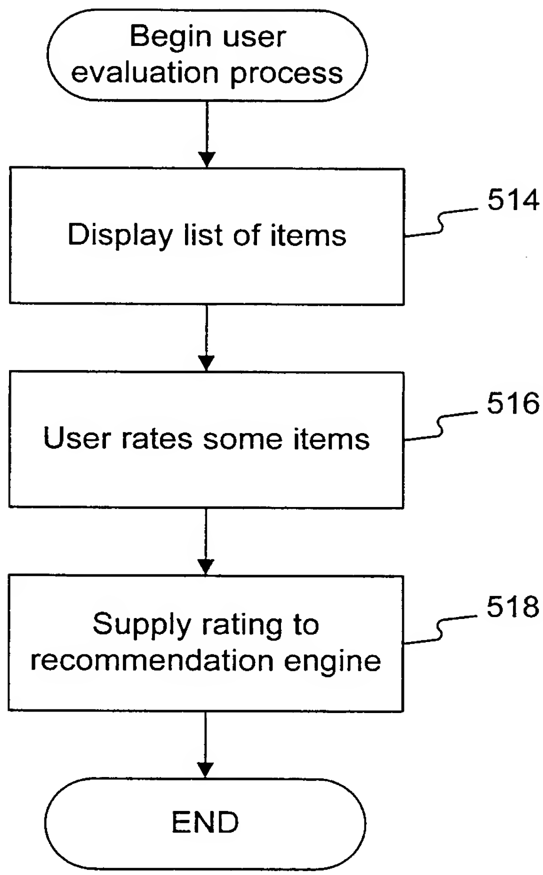

Begin user evaluation process
Display list of items
514
User rates some items
516
Supply rating to recommendation engine
518
END

Fig. 5B

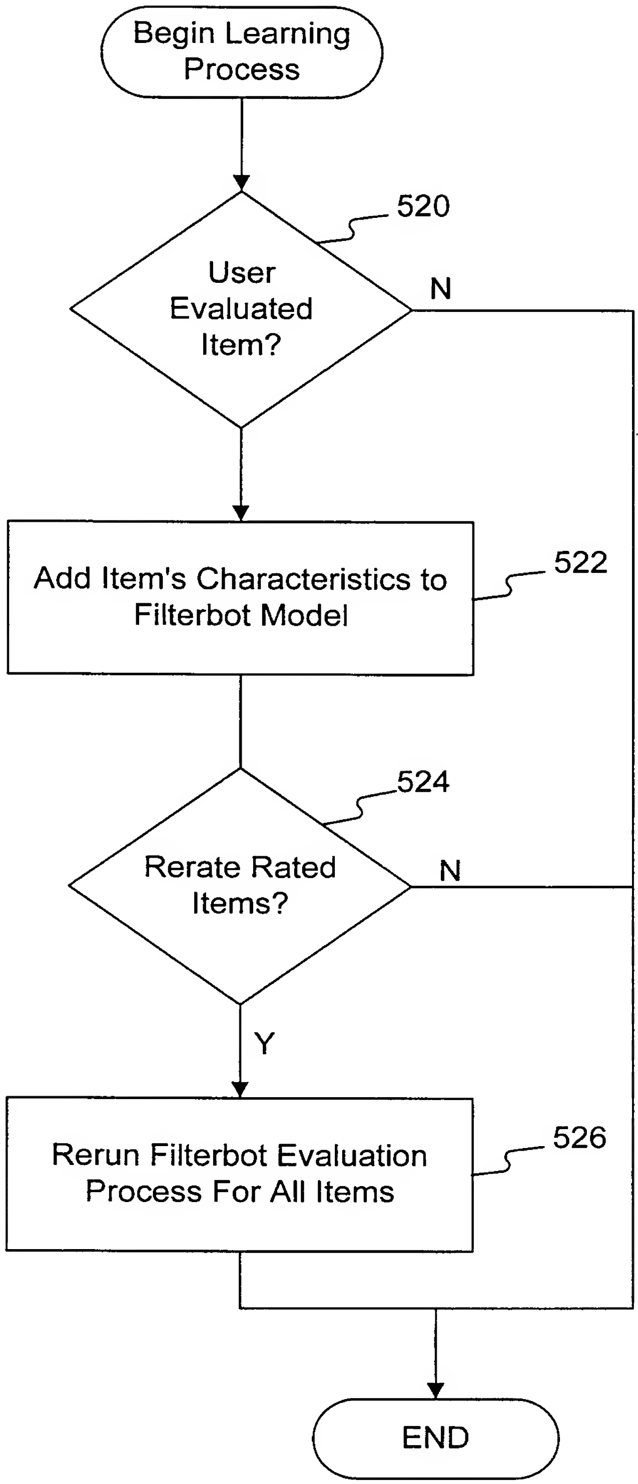

Begin Learning Process
520
User Evaluated Item?
N
Add Item's Characteristics to Filterbot Model
522
524
Rerate Rated Items?
N
Y
Rerun Filterbot Evaluation Process For All Items
526
END

Fig. 5C

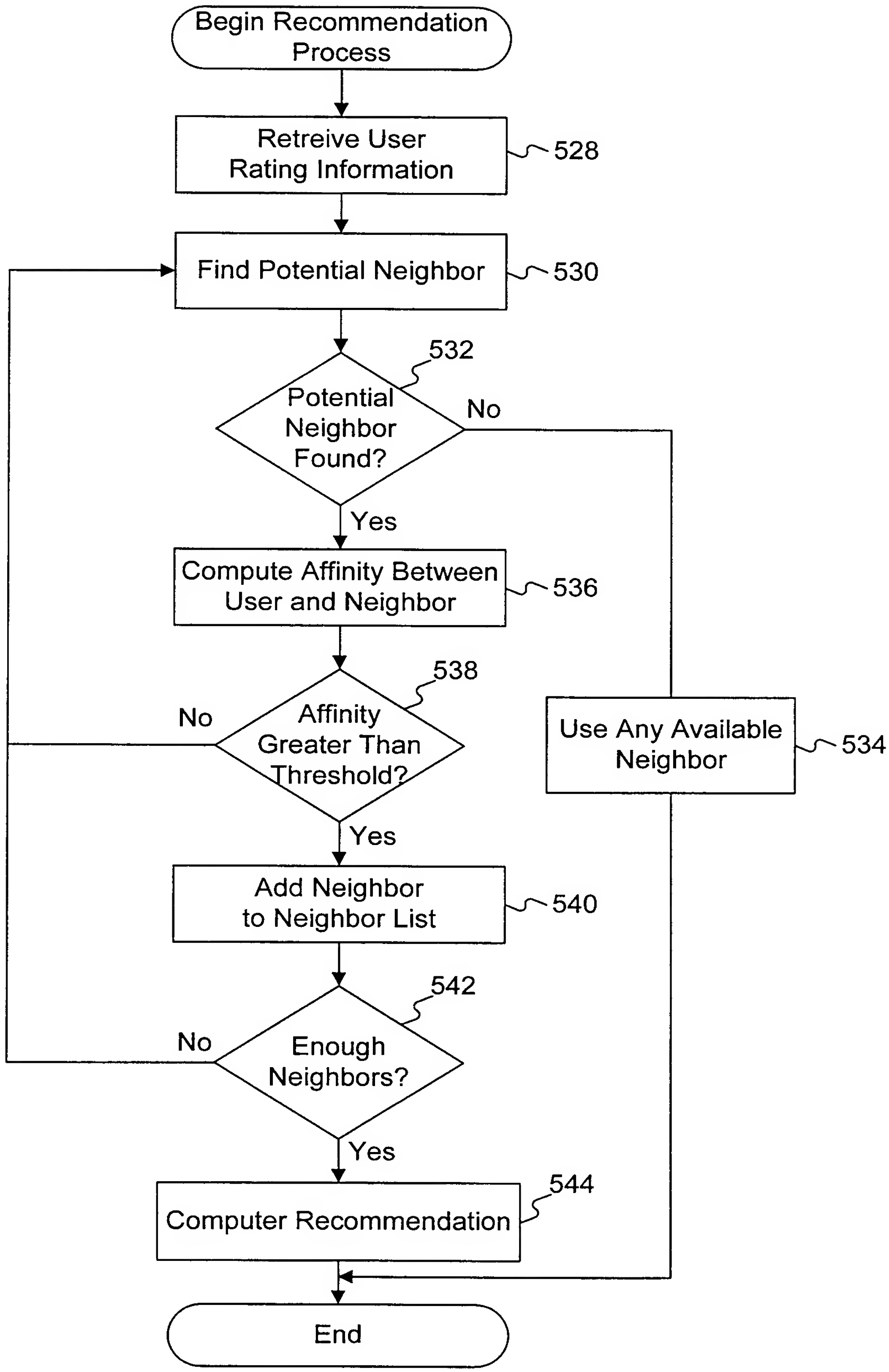

Begin Recommendation Process
Retreive User Rating Information
528
Find Potential Neighbor
530
532
Potential Neighbor Found?
No
Yes
Compute Affinity Between User and Neighbor
536
538
Affinity Greater Than Threshold?
No
Yes
Use Any Available Neighbor
534
Add Neighbor to Neighbor List
540
542
Enough Neighbors?
No
Yes
544
Computer Recommendation
End

FIG. 5D

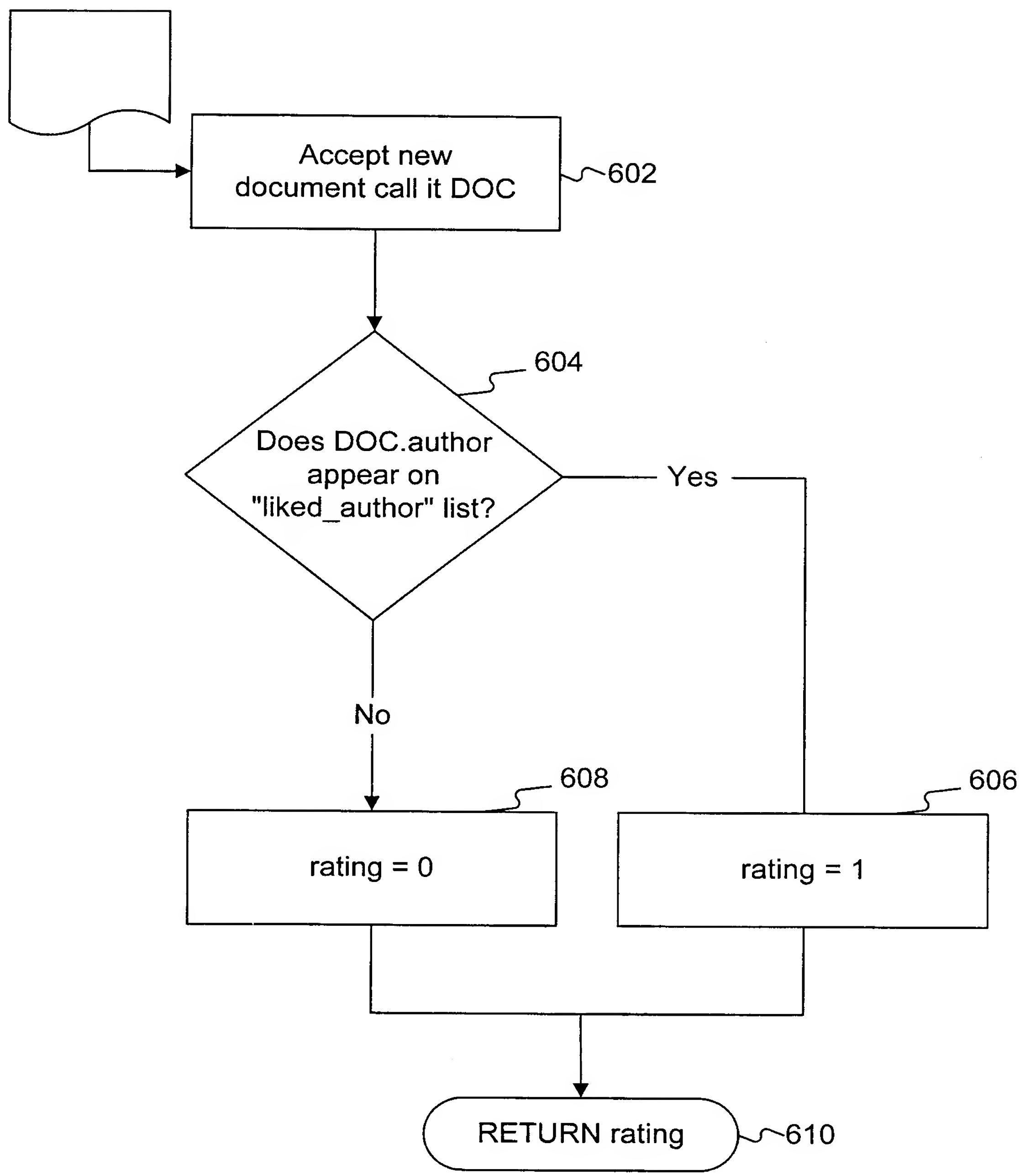

Accept new
document call it DOC
602
604
Does DOC.author
appear on
"liked_author" list?
Yes
No
608
606
rating = 0
rating = 1
RETURN rating
610

Fig. 6

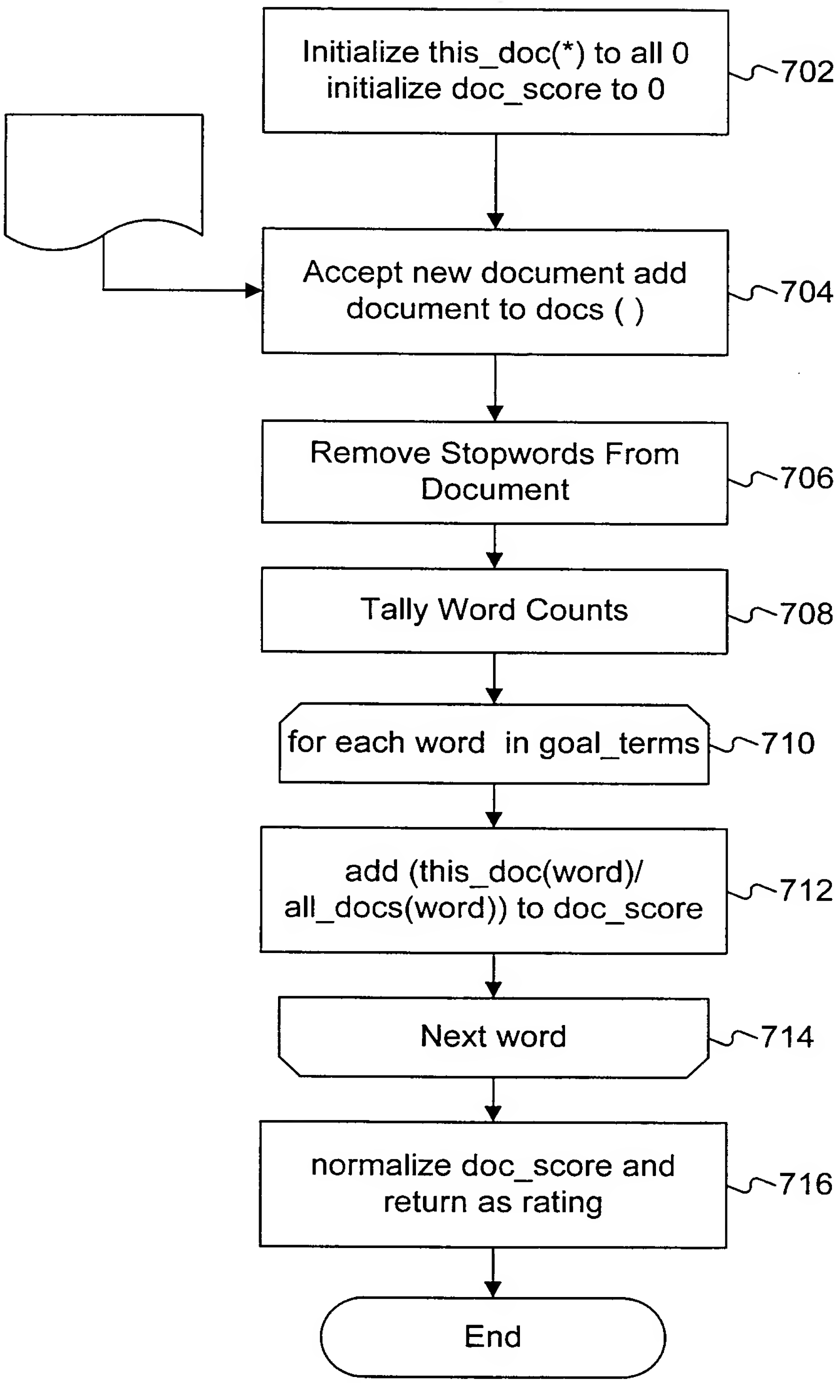

Initialize this_doc(*) to all 0
initialize doc_score to 0
702
Accept new document add
document to docs ( )
704
Remove Stopwords From
Document
706
Tally Word Counts
708
for each word in goal_terms
710
add (this_doc(word)/
all_docs(word)) to doc_score
712
Next word
714
normalize doc_score and
return as rating
716
End

Fig. 7

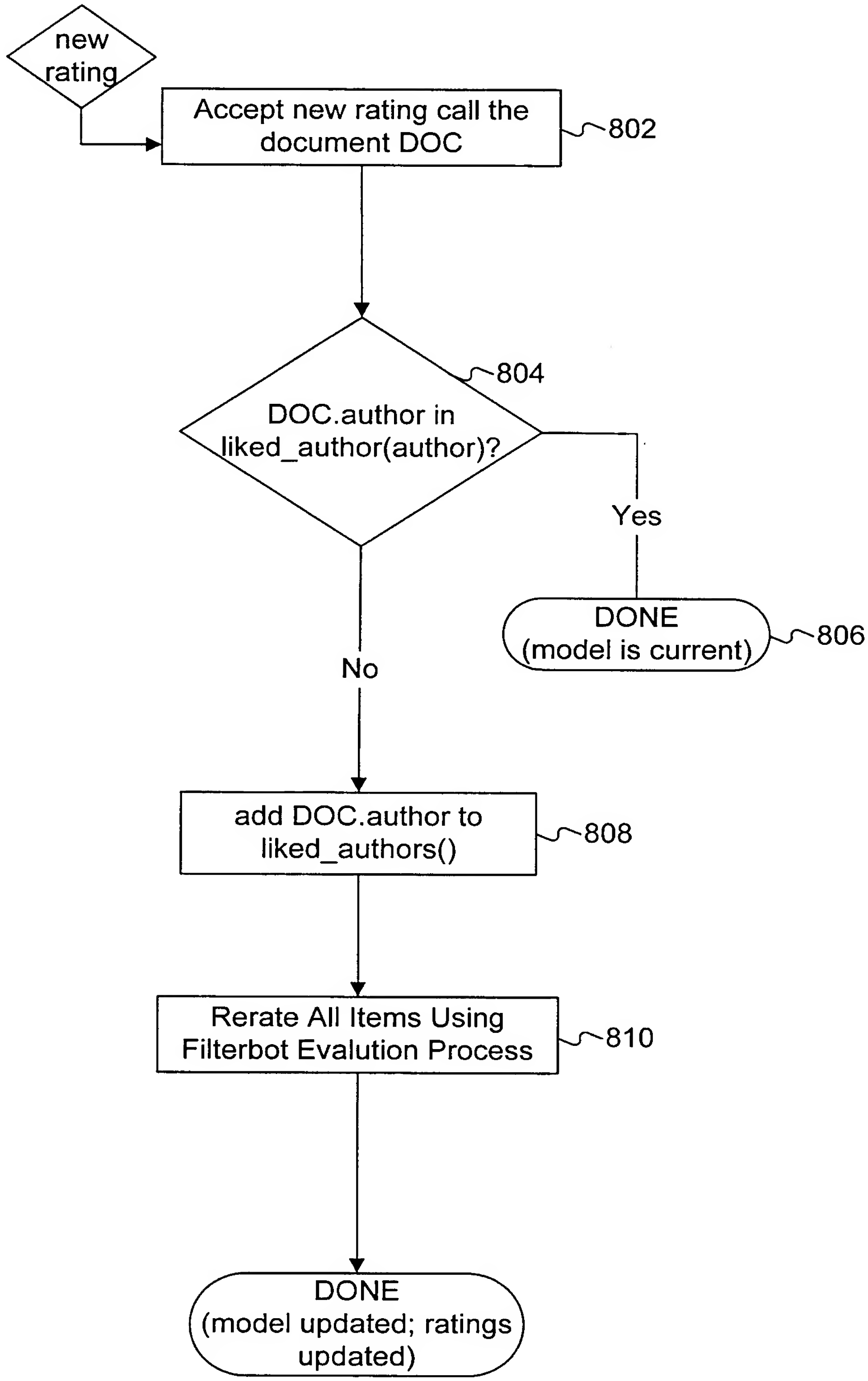

new rating
Accept new rating call the document DOC
802
DOC.author in liked_author(author)?
804
Yes
DONE (model is current)
806
No
add DOC.author to liked_authors()
808
Rerate All Items Using Filterbot Evalution Process
810
DONE (model updated; ratings updated)

Fig. 8

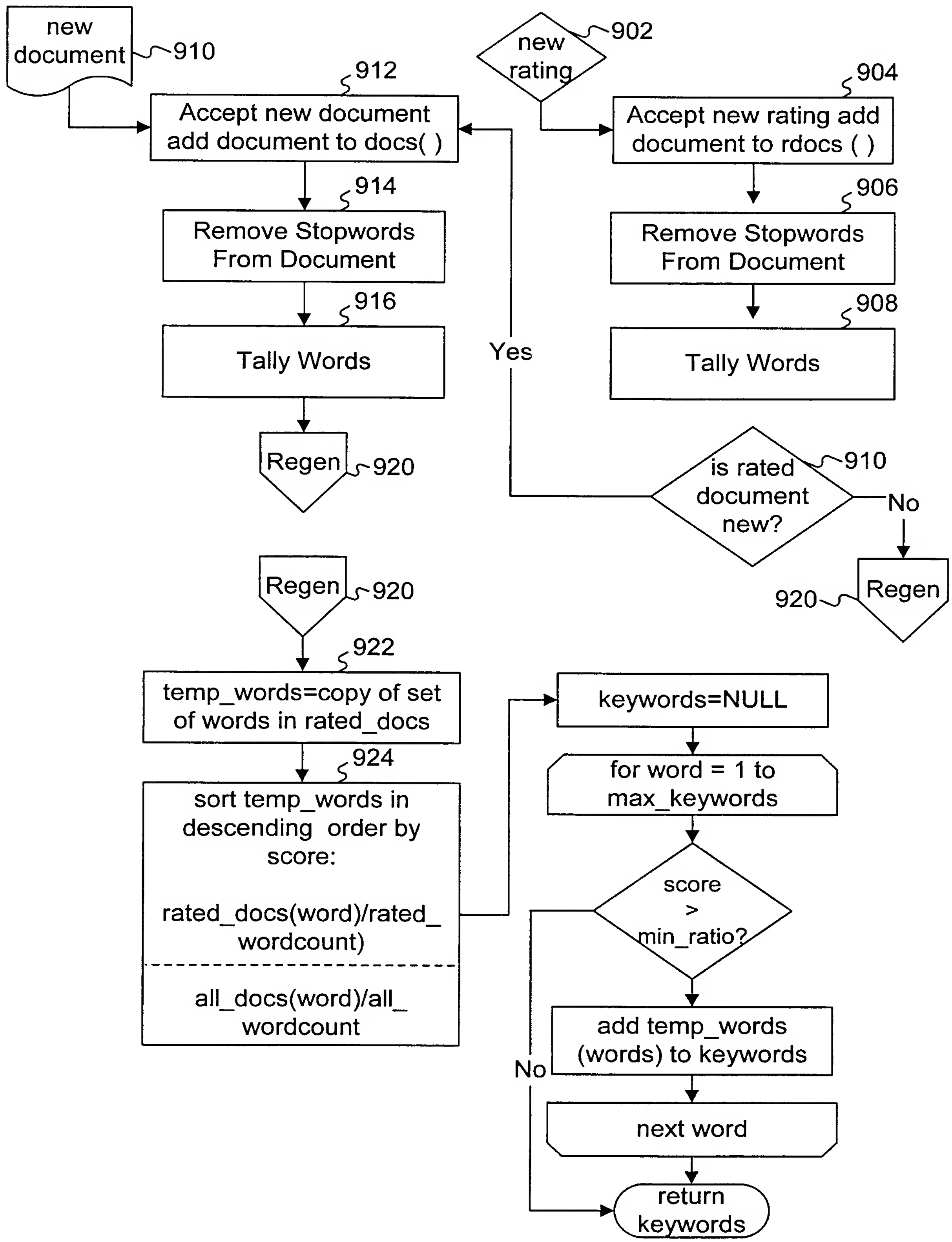

new document
910
912
Accept new document add document to docs( )
914
Remove Stopwords From Document
916
Tally Words
Regen
920
new rating
902
904
Accept new rating add document to rdocs ( )
906
Remove Stopwords From Document
908
Tally Words
Yes
is rated document new?
910
No
Regen
920
Regen
920
922
temp_words=copy of set of words in rated_docs
924
sort temp_words in descending order by score:
rated_docs(word)/rated_wordcount)
all_docs(word)/all_wordcount
keywords=NULL
for word = 1 to max_keywords
score > min_ratio?
add temp_words (words) to keywords
No
next word
return keywords

Fig. 9

INTERNATIONAL SEARCH REPORT	International application No. PCT/US00/28002

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :GO6F 19/00
US CL :705/7, 8

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : GO6F 17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST 2.0, CAS ONLINE, DIALOG, IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4,996,642 A (HEY) 26 February 1991, see entire document	1-68
A	US 4,775,935 A (YOURICK) 04 October 1988, see entire document.	1-68
A	US 5,749,081 A (WHITEIS) 05 May 1998, see entire document.	1-68
A	US 5,790,426 A (ROBINSON) 04 AUGUST 1998, see entire document.	1-68

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date		
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 DECEMBER 2000	09.01.2001

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	TARIQ R. HAFIZ James R. Matthews
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9643